

Distributed Bayesian Filtering using Logarithmic Opinion Pool for Dynamic Sensor Networks [★]

Saptarshi Bandyopadhyay ^a, Soon-Jo Chung ^{b,1},

^a*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA*

^b*Graduate Aerospace Laboratories, California Institute of Technology, Pasadena, CA 91125, USA*

Abstract

The discrete-time Distributed Bayesian Filtering (DBF) algorithm is presented for the problem of tracking a target dynamic model using a time-varying network of heterogeneous sensing agents. In the DBF algorithm, the sensing agents combine their normalized likelihood functions in a distributed manner using the logarithmic opinion pool and the dynamic average consensus algorithm. We show that each agent's estimated likelihood function globally exponentially converges to an error ball centered on the joint likelihood function of the centralized multi-sensor Bayesian filtering algorithm. We rigorously characterize the convergence, stability, and robustness properties of the DBF algorithm. Moreover, we provide an explicit bound on the time step size of the DBF algorithm that depends on the time-scale of the target dynamics, the desired convergence error bound, and the modeling and communication error bounds. Furthermore, the DBF algorithm for linear-Gaussian models is cast into a modified form of the Kalman information filter. The performance and robust properties of the DBF algorithm are validated using numerical simulations.

Key words: Bayesian filtering, distributed estimation, sensor network, data fusion, logarithmic opinion pool.

1 Introduction

A network of time-varying, heterogeneous sensing agents could use a distributed estimation algorithm to estimate the states of the target dynamics in a distributed manner. Potential applications include environment and pollution monitoring, analyzing communication and social networks, and tracking mobile targets in air, land, water, and space. In this paper, we present a new, discrete-time distributed estimation algorithm based on the logarithmic opinion pool that guarantees bounded convergence to the Bayesian-optimal probability distribution of the states of the target dynamics.

Discrete-time distributed estimation algorithms can be broadly classified into three categories based on their representation of the states of the target dynamics. Algorithms in the first category only estimate the mean and the covariance matrix of the target's states [1,2,3,4,5,6,7]. These algorithms usually deal with linearized target dynamics and measurement models, and also neglect information captured by the higher-order moments of the estimated probability distribution of the target's states. The second category aims to reach an agreement across the sensor network over a discrete set of hypotheses about the states of the target [8,9,10]. Although these algorithms use the entire information in the estimated probability distribution of the target's states, they are only applicable in cases where the target's states can be represented by a discrete (finite) set of hypotheses. Therefore, these algorithms are not suitable for estimation over continuous domains.

The third category of algorithms estimates the posterior probability distribution of the states of the target [11,12,13,14,15,16,17,18]. This category forms the most general class of distributed estimation algorithms because these algorithms can be used for estimation over continuous domains, and can incorporate nonlinear target dynamics, heterogeneous nonlinear measure-

[★] S. Bandyopadhyay and S.-J. Chung were supported in part by the AFOSR grant (FA95501210193) and the NSF grant (IIS-1253758). This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. ©2017 California Institute of Technology. All rights reserved.

Email addresses:

Saptarshi.Bandyopadhyay@jpl.nasa.gov (Saptarshi Bandyopadhyay), sjchung@caltech.edu (Soon-Jo Chung).

¹ Corresponding author. Tel.: +1 626 395 6294.

ment models, and non-Gaussian uncertainties. These algorithms also use the entire information (i.e., not just the mean and the covariance matrix) in the estimated probability distribution of the target's states. In light of these advantages, this paper focuses on the development of a distributed estimation algorithm that belongs to this third category.

In third-category algorithms, the agents exchange their local probability distributions with their neighboring agents and combine them using fusion or diffusive coupling rules to estimate the aggregate probability distribution. Schemes for combining probability distributions in a distributed manner, like the Linear Opinion Pool (LinOP) and the Logarithmic Opinion Pool (LogOP), were first studied in the statistics literature [19,20,21,22,23]. The LogOP scheme is deemed ideal for this purpose because of its favorable properties (e.g., externally Bayesian) [22].

We now focus on distributed estimation algorithms that use the LogOP scheme. The first such algorithm is proposed in [11]. In particular, [12] generates information-theoretically-optimal weights for the LogOP scheme. Combining probability distributions within the exponential family (i.e., probability distributions that can be expressed as exponential functions) is discussed in [13,14]. In the distributed estimation algorithm presented in [16] as well as in our prior work [17,18], the agents combine their local posterior probability distributions using the consensus algorithm, where the multiple consensus loops within each time step are executed much faster than the original time steps of the Bayesian filter. Moreover, [16,17,18] show that each agent's estimated probability distribution of the target's states converges around the pdf that minimizes the sum of Kullback–Leibler (KL) divergences from all the posterior probability distributions of the target's states. Similar algorithms for combining local likelihood functions using the consensus algorithm are proposed in [14,15]. But the number of consensus loops within each estimator time step grows very fast with the number of agents due to the convergence properties of the consensus algorithm [24,25,26]. Hence, such algorithms are not feasible if the time-scale of the target dynamics is comparatively fast. This connection between the time-scale of the target dynamics and the time step size of the distributed estimation algorithm has not been explored in the literature.

If all the agents are perfectly connected by a complete communication graph (i.e., each agent could communicate instantaneously with every other agent without any loss of information in the communication links), then the agents can exchange their local likelihood functions and use the centralized multi-sensor Bayesian filtering algorithm to estimate the Bayesian-optimal posterior probability distribution of the target's states. An open question is how to design a distributed estimation algo-

rithm for a time-varying, heterogeneous sensor network on a communication graph that is much sparser than a complete graph so that each agent's estimate converges to this Bayesian-optimal posterior probability distribution of the target's states. Furthermore, we assume that the time-varying communication network topology is periodically strongly connected and each agent can only communicate once with its neighboring agents during each time instant.

In this paper, we present the Distributed Bayesian Filtering (DBF) algorithm to address this open question. During each time instant, the agents exchange their normalized likelihood functions with their neighboring agents only once and then combine them using our fusion rule. Our fusion rule for combining arbitrary probability distributions relies on the LogOP scheme and the dynamic average consensus algorithm [24,25,26,27]. We show that after finite time instants, the estimated likelihood function of each agent converges to an error ball centered on the joint likelihood function of the centralized multi-sensor Bayesian filtering algorithm. We also provide an explicit upper bound on the time step size of the DBF algorithm that depends on the time-scale of the target dynamics and the convergence error bound. Moreover, we analyze the effect of communication and modeling errors on the DBF algorithm. If the target dynamics are linear-Gaussian models, we show that the DBF algorithm can be simplified to the modified (Kalman) information filter. Finally, we show that the distributed estimation algorithms in [14,15] are special cases of the DBF algorithm.

Furthermore, [16] analyzed their algorithm using linear-Gaussian models while [13] focused on probability distributions within the exponential family. In contrast, we present a rigorous proof technique, which was first introduced in our prior work [17,18], for the LogOP scheme that is applicable for general probability distributions.

This paper is organized as follows. Section 2 presents some preliminaries and the problem statement. The LogOP scheme and some general convergence results are presented in Section 3. The DBF algorithm and its special cases are presented in Section 4. Results of numerical simulations are presented in Section 5 and the paper is concluded in Section 6.

2 Preliminaries and Problem Statement

Let \mathbb{N} and \mathbb{R} represent the sets of natural numbers (positive integers) and real numbers respectively. The state space of the target's states \mathcal{X} is a closed set in \mathbb{R}^{n_x} , where n_x is the dimension of the states of the target. Let \mathcal{X} be the Borel σ -algebra for \mathcal{X} . A probability space is defined by the three-tuple $\{\mathcal{X}, \mathcal{X}, \mathbb{P}\}$, where \mathbb{P} is a complete, σ -additive probability measure on all \mathcal{X} . Let $p(\mathbf{x}) = \frac{d\mathbb{P}(\mathbf{x})}{d\mu(\mathbf{x})}$ denote the Radon–Nikodým density

of the probability distribution $\mathbb{P}(\mathbf{x})$ with respect to a measure $\mu(\mathbf{x})$. If $\mathbf{x} \in \mathcal{X}$ is continuous and $\mu(\mathbf{x})$ is a Lebesgue measure, $p(\mathbf{x})$ is the probability density function (pdf) [28]. Therefore, the probability of an event $\mathcal{A} \in \mathcal{X}$ can be written as the Lebesgue–Stieltjes integral $\mathbb{P}(\mathcal{A}) = \int_{\mathcal{A}} p(\mathbf{x}) d\mu(\mathbf{x})$. In this paper, we only deal with the continuous case where the function $p(\cdot)$ represents the pdf and $\mu(\cdot)$ is the Lebesgue measure. Let $\Phi(\mathcal{X})$ represent the set of all pdfs over the state space \mathcal{X} . The L_1 distance and the KL divergence between the pdfs $\mathcal{P}, \mathcal{Q} \in \Phi(\mathcal{X})$ are denoted by:

$$D_{L_1}(\mathcal{P}, \mathcal{Q}) = \int_{\mathcal{X}} |\mathcal{P}(\mathbf{x}) - \mathcal{Q}(\mathbf{x})| d\mu(\mathbf{x}),$$

$$D_{\text{KL}}(\mathcal{P}||\mathcal{Q}) = \int_{\mathcal{X}} \mathcal{P}(\mathbf{x}) \log\left(\frac{\mathcal{P}(\mathbf{x})}{\mathcal{Q}(\mathbf{x})}\right) d\mu(\mathbf{x}).$$

In this paper, all the algorithms are presented in discrete time. Let Δ be the time step size between any two consecutive time instants. The time index is denoted by a right subscript. The agent index is denoted by a lower-case right superscript. Frequently used symbols are listed in the following table.

Table 1: List of frequently used symbols

Symbol	Definition
\mathcal{A}_k	Adjacency matrix
\mathbf{b}	Periodicity of the communication network
\mathcal{J}_k^i	Inclusive neighbors of the i^{th} agent
\mathcal{L}_k^i	Normalized likelihood function
\mathcal{L}_k^C	Normalized joint likelihood function
N	Number of agents in the network
\mathcal{S}_k^i	Prior pdf
\mathcal{T}_k^i	Estimated likelihood function
\mathcal{U}_k^i	Estimated KL-divergence-minimizing pdf
\mathcal{W}_k^i	Posterior pdf
$\mathcal{W}_k^{C,i}$	Centralized posterior pdf
\mathcal{X}	State space
\mathbf{x}_k	True states of the target
$\mathbf{x}_{k k-1}$	Predicted states of the target
$\mathbf{x}_{k k}$	Updated states of the target
\mathbf{y}_k^i	Measurement taken by the i^{th} agent
Δ	Time step size

2.1 Target Dynamics and Measurement Models

Let \mathbf{x}_k represent the true states of the target at the k^{th} time instant, where $\mathbf{x}_k \in \mathcal{X}$ for all $k \in \mathbb{N}$. The dynamics of the target in discrete time is given by:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k, \Delta), \forall k \in \mathbb{N}, \quad (1)$$

where $\mathbf{f}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_x}$ is a possibly nonlinear time-varying function of the state \mathbf{x}_k , the discretization time step size Δ , and an independent and identically distributed (i.i.d.) process noise \mathbf{w}_k , where n_w is the dimension of the process noise vector.

Consider a network of N heterogeneous sensing agents simultaneously tracking this target. Let \mathbf{y}_k^i represent the measurement taken by the i^{th} agent at the k^{th} time instant. The measurement model of the agents is given by:

$$\mathbf{y}_k^i = \mathbf{h}_k^i(\mathbf{x}_k, \mathbf{v}_k^i), \forall i \in \mathcal{V} = \{1, \dots, N\}, \forall k \in \mathbb{N}, \quad (2)$$

where $\mathbf{h}_k^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_{vi}} \rightarrow \mathbb{R}^{n_{yi}}$ is a possibly nonlinear time-varying function of the state \mathbf{x}_k and an i.i.d. measurement noise \mathbf{v}_k^i , where n_{yi} and n_{vi} are dimensions of the measurement and measurement noise vectors respectively. Note that the measurements are conditionally independent given the target's states. We assume that the target dynamics (1) and measurement models (2) are known.

2.2 Bayesian Filtering Algorithm

Each agent uses the Bayesian filtering algorithm to estimate the pdf of the states of the target [29,28]. Let $\mathbf{x}_{k|k-1}$ and $\mathbf{x}_{k|k}$ represent the predicted and updated states of the target at the k^{th} time instant. Let the pdfs $\mathcal{S}_k^i = p(\mathbf{x}_{k|k-1}) \in \Phi(\mathcal{X})$ and $\mathcal{W}_k^i = p(\mathbf{x}_{k|k}) = p(\mathbf{x}_{k|k-1}|\mathbf{y}_k^i) \in \Phi(\mathcal{X})$ denote the i^{th} agent's prior and posterior pdfs of the target's states at the k^{th} time instant.

During the prediction step, the prior pdf $\mathcal{S}_k^i = p(\mathbf{x}_{k|k-1})$ is obtained from the previous posterior pdf $\mathcal{W}_{k-1}^i = p(\mathbf{x}_{k-1|k-1})$ using the Chapman–Kolmogorov equation [28]:

$$\mathcal{S}_k^i = \int_{\mathcal{X}} p(\mathbf{x}_{k|k-1}|\mathbf{x}_{k-1|k-1}) \mathcal{W}_{k-1}^i d\mu(\mathbf{x}_{k-1|k-1}), \quad (3)$$

where the probabilistic model of the state evolution $p(\mathbf{x}_{k|k-1}|\mathbf{x}_{k-1|k-1})$ is obtained from the known target dynamics model (1). We assume that the prior pdf is available at the start of the estimation process.

The new measurement \mathbf{y}_k^i is used to compute the posterior pdf $\mathcal{W}_k^i = p(\mathbf{x}_{k|k}) = p(\mathbf{x}_{k|k-1}|\mathbf{y}_k^i)$ during the update step using the Bayes' rule [28]:

$$\mathcal{W}_k^i = \frac{p(\mathbf{y}_k^i|\mathbf{x}_{k|k-1}) \mathcal{S}_k^i}{\int_{\mathcal{X}} p(\mathbf{y}_k^i|\mathbf{x}_{k|k-1}) \mathcal{S}_k^i d\mu(\mathbf{x}_{k|k-1})}. \quad (4)$$

The likelihood function $p(\mathbf{y}_k^i|\mathbf{x}_{k|k-1})$ is obtained from the i^{th} agent's known measurement model (2). Let the

pdf $\mathcal{L}_k^i \in \Phi(\mathcal{X})$ represent the normalized likelihood function, i.e.,

$$\mathcal{L}_k^i = \frac{p(\mathbf{y}_k^i | \mathbf{x}_{k|k-1})}{\int_{\mathcal{X}} p(\mathbf{y}_k^i | \mathbf{x}_{k|k-1}) d\mu(\mathbf{x}_{k|k-1})}. \quad (5)$$

Therefore, (4) is equivalent to $\mathcal{W}_k^i = \frac{\mathcal{L}_k^i \mathcal{S}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k^i \mathcal{S}_k^i d\mu(\mathbf{x}_{k|k-1})}$.

If all the sensing agents are hypothetically connected by a complete graph, then the agents can exchange their likelihood functions. Each agent can use the centralized multi-sensor Bayesian filtering algorithm to compute the centralized posterior pdf of the target's states $\mathcal{W}_k^{C,i} = p(\mathbf{x}_{k|k}) = p(\mathbf{x}_{k|k-1} | \mathbf{y}_k^1, \dots, \mathbf{y}_k^N) \in \Phi(\mathcal{X})$ using the Bayes' rule [30]:

$$\mathcal{W}_k^{C,i} = \frac{\mathcal{L}_k^C \mathcal{S}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k^C \mathcal{S}_k^i d\mu(\mathbf{x}_{k|k-1})}. \quad (6)$$

$$\text{where } \mathcal{L}_k^C = \frac{\prod_{j=1}^N \mathcal{L}_k^j}{\int_{\mathcal{X}} \prod_{j=1}^N \mathcal{L}_k^j d\mu(\mathbf{x}_{k|k-1})}. \quad (7)$$

Here, \mathcal{L}_k^C (7) is the normalized joint likelihood function.

Bayesian filtering is optimal because this posterior pdf $\mathcal{W}_k^{C,i}$ integrates and uses all the available information expressed by probabilities [28]. Moreover, an optimal state estimate with respect to any criterion can be computed from this posterior pdf $\mathcal{W}_k^{C,i}$. The minimum mean-square error (MMSE) estimate and the maximum a posteriori (MAP) estimate are given by [31]:

$$\hat{\mathbf{x}}_{k|k}^{MMSE} = \int_{\mathcal{X}} \mathbf{x} \mathcal{W}_k^{C,i} d\mu(\mathbf{x}), \quad \hat{\mathbf{x}}_{k|k}^{MAP} = \arg \max_{\mathbf{x} \in \mathcal{X}} \mathcal{W}_k^{C,i}.$$

Other potential criteria for optimality, such as maximum likelihood, minimum conditional KL divergence, and minimum free energy, are discussed in [28,31]. The main advantage of the original Bayesian filtering formulation is that no approximation is needed during the filtering process; i.e., the complete information about the dynamics and uncertainties of the model can be incorporated in the filtering algorithm. However, direct implementation of Bayesian filtering (3)–(4) is computationally expensive. Practical implementation of these algorithms, in their most general form, is achieved using particle filtering [29,32] and Bayesian programming [33,34].

2.3 Problem Statement

Let the pdf $\mathcal{T}_k^i \in \Phi(\mathcal{X})$ denote the estimated joint likelihood function of the i^{th} agent at the k^{th} time instant. Let $\eta \in (0, 1)$ and $\delta \in (0, \frac{2}{1+\eta})$ be positive constants. Our aim is to design a discrete-time distributed estimation algorithm, over the communication network topology described in Section 2.4, so that each agent's \mathcal{T}_k^i converges to the normalized joint likelihood function \mathcal{L}_k^C (7),

where the convergence error is given by

$$D_{L_1}(\mathcal{T}_k^i, \mathcal{L}_k^C) \leq (1 + \eta)\delta, \quad \forall k \geq \kappa, \forall i \in \mathcal{V}, \quad (8)$$

$$\lim_{k \rightarrow \infty} D_{L_1}(\mathcal{T}_k^i, \mathcal{L}_k^C) \leq \delta, \quad \forall i \in \mathcal{V}. \quad (9)$$

The DBF algorithm, shown in Fig. 1 and Algorithm 1, achieves this objective. Note that the agents exchange their estimated pdfs with their neighboring agents only once during each time instant before the fusion step.

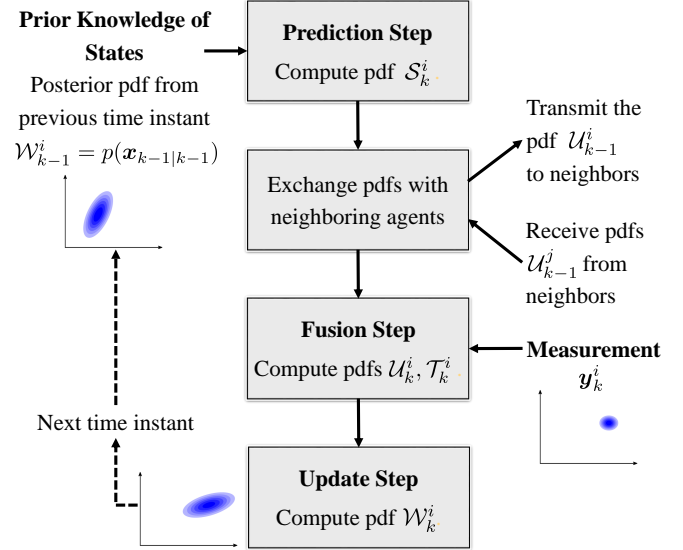


Fig. 1. Flowchart of the DBF algorithm (for the i^{th} agent at the k^{th} time instant)

2.4 Communication Network Topology

The time-varying communication network topology of the sensor network is denoted by the directed graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$. The edge $(i, j) \in \mathcal{E}_k$ if and only if the i^{th} agent receives information from the j^{th} agent at the k^{th} time instant. The inclusive neighbors of the i^{th} agent are denoted by $\mathcal{J}_k^i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}_k\} \cup \{i\}$. The matrix $\mathcal{A}_k \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of \mathcal{G}_k , where $\mathcal{A}_k[i, j] \neq 0$ if and only if $j \in \mathcal{J}_k^i$.

Assumption 1 [26,27] The digraph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ and its adjacency matrix \mathcal{A}_k satisfy the following properties:

- (i) There exists some positive integer $\mathbf{b} \in \mathbb{N}$ such that the directed graph $(\mathcal{V}, \mathcal{E}_k \cup \mathcal{E}_{k+1} \cup \dots \cup \mathcal{E}_{k+\mathbf{b}-1})$ is strongly connected for all time instants $k \in \mathbb{N}$.
- (ii) The matrix \mathcal{A}_k is doubly stochastic, i.e., $\mathbf{1}^T \mathcal{A}_k = \mathbf{1}^T$ and $\mathcal{A}_k \mathbf{1} = \mathbf{1}$ for all $k \in \mathbb{N}$, where $\mathbf{1} = [1, 1, \dots, 1]^T$.
- (iii) The matrix product $\mathcal{A}_{k, k+\mathbf{b}-1}$ is defined as:

$$\mathcal{A}_{k, k+\mathbf{b}-1} = \left(\prod_{\tau=k}^{k+\mathbf{b}-1} \mathcal{A}_\tau \right). \quad (10)$$

There exists a constant $\gamma \in (0, \frac{1}{2})$ such that each element $\mathcal{A}_{k, k+\mathbf{b}-1}[i, j] \in [\gamma, 1] \cup \{0\}$ for all $i, j \in \mathcal{V}$ and $k \in \mathbb{N}$.

Therefore, the digraph \mathcal{G}_k is periodically strongly connected and the matrix \mathcal{A}_k is non-degenerate and balanced.

Note that if $\mathbf{b} = 1$, then the digraph \mathcal{G}_k is strongly connected at all time instants $k \in \mathbb{N}$.

3 Logarithmic Opinion Pool and Convergence Results

Let the pdf $\mathcal{P}_k^i \in \Phi(\mathcal{X})$ denote the i^{th} agent's pdf at the k^{th} time instant. The LinOP and LogOP schemes for combining the pdfs \mathcal{P}_k^i are given by [20]:

$$\mathcal{P}_k^{\text{LinOP}}(\mathbf{x}) = \sum_{i=1}^N \alpha_k^i \mathcal{P}_k^i(\mathbf{x}), \quad (11)$$

$$\mathcal{P}_k^{\text{LogOP}}(\mathbf{x}) = \frac{\prod_{i=1}^N (\mathcal{P}_k^i(\mathbf{x}))^{\alpha_k^i}}{\int_{\mathcal{X}} \prod_{i=1}^N (\mathcal{P}_k^i(\tilde{\mathbf{x}}))^{\alpha_k^i} d\mu(\tilde{\mathbf{x}})}, \quad (12)$$

where the weights α_k^i are such that $\sum_{i=1}^N \alpha_k^i = 1$ and the integral in the denominator of (12) is finite. Thus, the combined pdf obtained using LinOP and LogOP gives the weighted algebraic and geometric averages of the individual pdfs respectively. As shown in Fig. 2, the combined pdf obtained using LogOP typically preserves the multimodal or unimodal nature of the original individual pdfs [22]. The most compelling reason for using the LogOP scheme is that it is externally Bayesian; i.e., the LogOP combination step commutes with the process of updating the pdfs by multiplying with a commonly agreed likelihood pdf $\mathcal{L}_k \in \Phi(\mathcal{X})$:

$$\frac{\mathcal{L}_k \mathcal{P}_k^{\text{LogOP}}}{\int_{\mathcal{X}} \mathcal{L}_k \mathcal{P}_k^{\text{LogOP}} d\mu(\tilde{\mathbf{x}})} = \frac{\prod_{i=1}^N \left(\frac{\mathcal{L}_k \mathcal{P}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k \mathcal{P}_k^i d\mu(\tilde{\mathbf{x}})} \right)^{\alpha_k^i}}{\int_{\mathcal{X}} \prod_{i=1}^N \left(\frac{\mathcal{L}_k \mathcal{P}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k \mathcal{P}_k^i d\mu(\tilde{\mathbf{x}})} \right)^{\alpha_k^i} d\mu(\tilde{\mathbf{x}})}.$$

Therefore, the LogOP scheme is ideal for combining pdfs in distributed estimation algorithms.

Due to the multiplicative nature of the LogOP scheme, each agent has veto power [22]. That is, if $\mathcal{P}_k^i(\mathbf{x}) = 0$ for some $\mathbf{x} \in \mathcal{X}$ and some agent $i \in \mathcal{V}$ with $\alpha_k^i > 0$, then $\mathcal{P}_k^{\text{LogOP}}(\mathbf{x}) = 0$ in the combined pdf irrespective of the pdfs of the other agents. In order to avoid this veto condition, we enforce the following assumption which has been used in the literature.

Assumption 2 [10,22] (Nonzero Probability Property) *In this paper, all pdfs are strictly positive everywhere in the closed set \mathcal{X} .*

In order to analyze the LogOP scheme with general probability distributions that satisfy Assumption 2, we use the following functions.

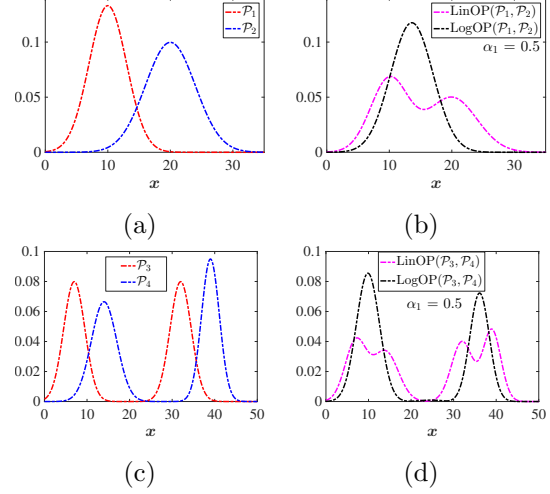


Fig. 2. The pdfs in (a) and (c) are combined using LinOP and LogOP in (b) and (d). Note that the LogOP solution preserves the modal nature of the original pdfs.

Definition 1 Under Assumption 2, for any constant $\psi \in \mathcal{X}$, we have $\mathcal{P}_k^i(\psi) > 0, \forall i \in \mathcal{V}$ and $\mathcal{P}_k^{\text{LogOP}}(\psi) > 0$. Using simple algebraic manipulation of (12), we get [23]:

$$\mathcal{P}_k^{\text{LogOP}}(\mathbf{x}) := \ln \left[\frac{\mathcal{P}_k^{\text{LogOP}}(\mathbf{x})}{\mathcal{P}_k^{\text{LogOP}}(\psi)} \right] = \sum_{i=1}^N \alpha_k^i \mathcal{P}_k^i(\mathbf{x}), \quad (13)$$

$$\text{where } \mathcal{P}_k^i(\mathbf{x}) := \ln \left[\frac{\mathcal{P}_k^i(\mathbf{x})}{\mathcal{P}_k^i(\psi)} \right], \forall i \in \mathcal{V}. \quad (14)$$

Thus, we have represented the LogOP scheme (12) as a linear equation using these functions $\mathcal{P}_k^i(\mathbf{x})$ and $\mathcal{P}_k^{\text{LogOP}}(\mathbf{x})$, and removed the effect of the normalizing constants.

We now state some useful convergence results using the functions in Definition 1. The proofs are given in Appendix.

Definition 2 (Pointwise Convergence) *The pdf \mathcal{P}_k^i converges pointwise to the pdf $\mathcal{P}^* \in \Phi(\mathcal{X})$, if and only if $\lim_{k \rightarrow \infty} \mathcal{P}_k^i(\mathbf{x}) = \mathcal{P}^*(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$.*

Lemma 1 *If the pdfs \mathcal{P}, \mathcal{Q} satisfy Assumption 2, then there exists $\psi \in \mathcal{X}$ such that $\mathcal{P}(\psi) = \mathcal{Q}(\psi)$.*

Lemma 2 *If the function \mathcal{P}_k^i (14) converges pointwise to the function $\mathcal{P}^* := \ln \left[\frac{\mathcal{P}^*(\mathbf{x})}{\mathcal{P}^*(\psi)} \right]$, then the corresponding pdf \mathcal{P}_k^i also converges pointwise to the pdf \mathcal{P}^* .*

Definition 3 (Convergence in TV) *The measure $\mu_{\mathcal{P}_k^i}$ is defined as the measure induced by the pdf \mathcal{P}_k^i on \mathcal{X} , where $\mu_{\mathcal{P}_k^i}(\mathcal{A}) = \int_{\mathcal{A}} \mathcal{P}_k^i d\mu(\mathbf{x})$ for any event $\mathcal{A} \in \mathcal{X}$. Similarly, let $\mu_{\mathcal{P}^*}$ denote the measure induced by the pdf \mathcal{P}^* on \mathcal{X} .*

The TV distance is defined as $\|\mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^*}\|_{\text{TV}} := \sup_{\mathcal{A} \in \mathcal{X}} |\mu_{\mathcal{P}_k^i}(\mathcal{A}) - \mu_{\mathcal{P}^*}(\mathcal{A})|$. The measure $\mu_{\mathcal{P}_k^i}$ converges to the measure $\mu_{\mathcal{P}^*}$ in TV, if and only if $\|\lim_{k \rightarrow \infty} \mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^*}\|_{\text{TV}} = 0$.

Lemma 3 *If the pdf \mathcal{P}_k^i converges pointwise to the pdf \mathcal{P}^* , then the measure $\mu_{\mathcal{P}_k^i}$ converges in TV to the measure $\mu_{\mathcal{P}^*}$. Moreover, $\|\mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^*}\|_{\text{TV}} = \frac{1}{2} D_{L_1}(\mathcal{P}_k^i, \mathcal{P}^*)$.*

Another reason for using the LogOP scheme is that it minimizes the information lost during the combination process, where the information loss is measured using the KL divergence.

Lemma 4 [16,17] *The pdf $\mathcal{P}_k^{\text{KL}} \in \Phi(\mathcal{X})$ that globally minimizes the sum of KL divergences with the pdfs \mathcal{P}_k^i for all agents is given by:*

$$\mathcal{P}_k^{\text{KL}} = \arg \min_{\rho \in \Phi(\mathcal{X})} \sum_{i=1}^N D_{\text{KL}}(\rho \| \mathcal{P}_k^i) = \frac{\prod_{i=1}^N (\mathcal{P}_k^i)^{\frac{1}{N}}}{\int_{\mathcal{X}} \prod_{i=1}^N (\mathcal{P}_k^i)^{\frac{1}{N}} d\mu(\bar{\mathbf{x}})}.$$

Note that the pdf $\mathcal{P}_k^{\text{KL}}$ is equivalent to the pdf $\mathcal{P}_k^{\text{LogOP}}$ (12) obtained using the LogOP scheme with weights $\alpha_k^i = \frac{1}{N}$ for all agents.

The proof of Lemma 4 is given in our prior work [17]. Note that the normalized joint likelihood function \mathcal{L}_k^C (7) is also given by:

$$\mathcal{L}_k^C = \frac{\prod_{j=1}^N \mathcal{L}_k^j}{\int_{\mathcal{X}} \prod_{j=1}^N \mathcal{L}_k^j d\mu(\bar{\mathbf{x}})} = \frac{(\mathcal{L}_k^{\text{KL}})^N}{\int_{\mathcal{X}} (\mathcal{L}_k^{\text{KL}})^N d\mu(\bar{\mathbf{x}})}, \quad (15)$$

$$\text{where } \mathcal{L}_k^{\text{KL}} = \frac{\prod_{j=1}^N (\mathcal{L}_k^j)^{\frac{1}{N}}}{\int_{\mathcal{X}} \prod_{j=1}^N (\mathcal{L}_k^j)^{\frac{1}{N}} d\mu(\bar{\mathbf{x}})}. \quad (16)$$

We show that the DBF algorithm also estimates the pdf $\mathcal{L}_k^{\text{KL}}$ (16) in a distributed manner.

4 Distributed Bayesian Filtering Algorithm

In this section, we present the main DBF algorithm, its convergence and robustness properties, and its application to special cases. We first state an assumption on the time-varying nature of the pdfs \mathcal{L}_k^i (5) for all agents that directly link the target dynamics and measurement models with the time step size of the distributed estimation algorithm.

Assumption 3 *For any time step size $\Delta > 0$, there exists a time-invariant constant $\theta_L > 0$ such that for all agents $i \in \mathcal{V} = \{1, \dots, N\}$:*

$$e^{-\Delta\theta_L} \leq \frac{\mathcal{L}_k^i(\mathbf{x})}{\mathcal{L}_{k-1}^i(\mathbf{x})} \leq e^{\Delta\theta_L}, \quad \forall \mathbf{x} \in \mathcal{X}, \quad \forall k \in \mathbb{N}. \quad (17)$$

The necessary conditions for satisfying (17) are given by $D_{\text{KL}}(\mathcal{L}_k^i \| \mathcal{L}_{k-1}^i) \leq \Delta\theta_L$ and $D_{\text{KL}}(\mathcal{L}_{k-1}^i \| \mathcal{L}_k^i) \leq \Delta\theta_L$.

We now state the DBF algorithm, whose steps are shown in Fig. 1. Let the pdf $\mathcal{U}_k^i \in \Phi(\mathcal{X})$ denote the estimated KL-divergence-minimizing pdf of the i^{th} agent at the k^{th} time instant. The pdf \mathcal{T}_k^i is defined in Section 2.3. Under Assumptions 1–3, the pseudo-code of the DBF algorithm is given in Algorithm 1.

Algorithm 1. Distributed Bayesian Filtering Algorithm

1. (i^{th} agent's steps at k^{th} time instant)
2. Compute prior pdf $\mathcal{S}_k^i = p(\mathbf{x}_{k|k-1})$ using (3)
3. Obtain local measurement \mathbf{y}_k^i
4. Compute normalized likelihood function \mathcal{L}_k^i using (5)
5. Receive pdfs \mathcal{U}_{k-1}^j from agents $j \in \mathcal{J}_k^i$
6. Compute pdfs \mathcal{U}_k^i and \mathcal{T}_k^i as follows

$$\mathcal{U}_k^i = \begin{cases} \mathcal{L}_1^i & \text{if } k = 1 \\ \frac{\Lambda \mathcal{L}_k^i (\mathcal{L}_{k-1}^i)^{-1}}{\int_{\mathcal{X}} \Lambda \mathcal{L}_k^i (\mathcal{L}_{k-1}^i)^{-1} d\mu(\bar{\mathbf{x}})} & \text{if } k \geq 2 \end{cases}, \quad (18)$$

$$\text{where } \Lambda = \prod_{j \in \mathcal{J}_k^i} (\mathcal{U}_{k-1}^j)^{\mathcal{A}_k^{[i,j]}},$$

$$\mathcal{T}_k^i = \frac{(\mathcal{U}_k^i)^N}{\int_{\mathcal{X}} (\mathcal{U}_k^i)^N d\mu(\bar{\mathbf{x}})}, \quad (19)$$

7. Compute posterior pdf $\mathcal{W}_k^i = p(\mathbf{x}_{k|k})$ as follows

$$\mathcal{W}_k^i = p(\mathbf{x}_{k|k}) = \frac{\mathcal{T}_k^i \mathcal{S}_k^i}{\int_{\mathcal{X}} \mathcal{T}_k^i \mathcal{S}_k^i d\mu(\bar{\mathbf{x}})}. \quad (20)$$

The following theorem shows that the DBF algorithm satisfies the problem statement (8)–(9) in Section 2.3.

Theorem 5 *Under Assumptions 1–3, if all the agents execute the DBF algorithm (Algorithm 1) and the time step size Δ for (17) for Algorithm 1 is defined as*

$$\Delta = \frac{(1 - \sigma_m) \log(\delta + 1)}{2bN(N-1)\sqrt{N}\theta_L}, \quad (21)$$

then the steady-state convergence error between the pdf \mathcal{T}_k^i (19) and the pdf \mathcal{L}_k^C (7) is bounded by δ :

$$\lim_{k \rightarrow \infty} \max_{i \in \mathcal{V} = \{1, \dots, N\}} D_{L_1}(\mathcal{T}_k^i, \mathcal{L}_k^C) \leq \delta. \quad (22)$$

Furthermore, the convergence error between the pdfs \mathcal{T}_k^i (19) and \mathcal{L}_k^C (7) after κ time instants is bounded by $(1 + \eta)\delta$:

$$\max_{i \in \mathcal{V} = \{1, \dots, N\}} D_{L_1}(\mathcal{T}_k^i, \mathcal{L}_k^C) \leq (1 + \eta)\delta, \quad \forall k \geq \kappa, \quad (23)$$

where, $\kappa = 1$ if $\mathfrak{D}_1 \leq \frac{\log(\delta+1)}{N^{\frac{3}{2}}}$, otherwise:

$$\kappa = \left\lceil \frac{\mathbf{b}(N-1)}{\log \sigma_m} \log \left(\frac{\log \left(\frac{(1+\eta)\delta+1}{\delta+1} \right)}{\log \left(\frac{e^{N\frac{3}{2}\mathfrak{D}_1}}{\delta+1} \right)} \right) \right\rceil + 1, \quad (24)$$

$$\mathfrak{D}_1 = 2 \ln \left(\max_{\ell, j \in \mathcal{V}} \max_{\mathbf{x} \in \mathcal{X}} \frac{\mathcal{L}_1^\ell(\mathbf{x})}{\mathcal{L}_1^j(\mathbf{x})} \right). \quad (25)$$

Here, $\eta \in (0, 1)$ and $\delta \in (0, \frac{2}{1+\eta})$ are positive constants, and $\sigma_m = \max_{k \in \mathbb{N}} \sigma_{N-1}(\mathcal{A}_{k,k+\mathbf{b}(N-1)-1})$, where σ_{N-1} denotes the second largest singular value of the matrix, and σ_m is upper bounded by:

$$\sigma_m \leq \left(1 - \frac{4(\gamma - \gamma^N)}{(1 - \gamma)} \sin^2 \frac{\pi}{2N} \right)^{\frac{1}{2}} < 1. \quad (26)$$

Moreover, \mathbf{b} is the periodicity of the communication network topology, $\gamma \in (0, \frac{1}{2})$ is the smallest positive element in $\mathcal{A}_{k,k+\mathbf{b}-1}$ defined in Assumption 1, and θ_L is defined in Assumption 3.

The TV error between the measures induced by the pdfs \mathcal{T}_k^i and \mathcal{L}_k^C is bounded by:

$$\max_{i \in \mathcal{V}} \|\mu_{\mathcal{T}_k^i} - \mu_{\mathcal{L}_k^C}\|_{\text{TV}} \leq \frac{(1+\eta)\delta}{2}, \quad \forall k \geq \kappa, \quad (27)$$

$$\lim_{k \rightarrow \infty} \max_{i \in \mathcal{V}} \|\mu_{\mathcal{T}_k^i} - \mu_{\mathcal{L}_k^C}\|_{\text{TV}} \leq \frac{\delta}{2}. \quad (28)$$

Proof: Using Definition 1, we define the functions $\mathcal{L}_k^{\text{KL}}(\mathbf{x}) = \ln \left[\frac{\mathcal{L}_k^{\text{KL}}(\mathbf{x})}{\mathcal{L}_k^{\text{KL}}(\psi)} \right]$, $\mathcal{L}_k^C(\mathbf{x}) = \ln \left[\frac{\mathcal{L}_k^C(\mathbf{x})}{\mathcal{L}_k^C(\psi)} \right]$, $\mathcal{L}_k^i(\mathbf{x}) = \ln \left[\frac{\mathcal{L}_k^i(\mathbf{x})}{\mathcal{L}_k^i(\psi)} \right]$, $\mathcal{U}_k^i(\mathbf{x}) = \ln \left[\frac{\mathcal{U}_k^i(\mathbf{x})}{\mathcal{U}_k^i(\psi)} \right]$, and $\mathcal{T}_k^i(\mathbf{x}) = \ln \left[\frac{\mathcal{T}_k^i(\mathbf{x})}{\mathcal{T}_k^i(\psi)} \right]$ for all $i \in \mathcal{V}$. Since these functions are defined for all $\mathbf{x} \in \mathcal{X}$, we henceforth drop the term (\mathbf{x}) for brevity.

Step 1. We first show that the pdf \mathcal{U}_k^i (18) converges to the pdf $\mathcal{L}_k^{\text{KL}}$ (16). Equation (18) can be re-written using these functions as:

$$\mathcal{U}_k^i = \begin{cases} \mathcal{L}_1^i & \text{if } k = 1 \\ \sum_{j=1}^N \mathcal{A}_k[i, j] \mathcal{U}_{k-1}^j + \mathcal{L}_k^i - \mathcal{L}_{k-1}^i & \text{if } k \geq 2 \end{cases}, \quad (29)$$

because $\mathcal{A}_k[i, j] = 0$ if $j \notin \mathcal{J}_k^i$, as defined in Section 2.4. Since \mathcal{A}_k is doubly stochastic, (29) satisfies the conservation property:

$$\begin{aligned} \sum_{i=1}^N \mathcal{U}_k^i &= \sum_{i=1}^N \sum_{j=1}^N \mathcal{A}_k[i, j] \mathcal{U}_{k-1}^j + \sum_{i=1}^N (\mathcal{L}_k^i - \mathcal{L}_{k-1}^i), \\ &= \sum_{i=1}^N \left(\sum_{j=1}^N \mathcal{A}_k[j, i] \right) \mathcal{U}_{k-1}^i + \sum_{i=1}^N (\mathcal{L}_k^i - \mathcal{L}_{k-1}^i), \\ &= \sum_{i=1}^N (\mathcal{U}_1^i - \mathcal{L}_1^i) + \sum_{i=1}^N \mathcal{L}_k^i = N \mathcal{L}_k^{\text{KL}}. \end{aligned} \quad (30)$$

Note that $\mathcal{L}_k^{\text{KL}} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_k^i$ follows from (16). This shows that if the functions \mathcal{U}_k^i converge towards each other, then they will converge to the function $\mathcal{L}_k^{\text{KL}}$. Let us define the error vector \mathbf{e}_k as:

$$\mathbf{e}_k = \left[\mathcal{U}_k^1 - \mathcal{L}_k^{\text{KL}}, \dots, \mathcal{U}_k^i - \mathcal{L}_k^{\text{KL}}, \dots, \mathcal{U}_k^N - \mathcal{L}_k^{\text{KL}} \right]^T.$$

The evolution of the error vector \mathbf{e}_k is given by:

$$\mathbf{e}_k = \mathcal{A}_k \mathbf{e}_{k-1} + \mathbf{\Omega}_{k,k}, \quad \forall k \geq 2, \quad (31)$$

$$\text{where } \mathbf{\Omega}_{k,k} = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \begin{bmatrix} \mathcal{L}_k^1 - \mathcal{L}_{k-1}^1 \\ \vdots \\ \mathcal{L}_k^N - \mathcal{L}_{k-1}^N \end{bmatrix}.$$

The overall evolution of the error vector \mathbf{e}_k after $\mathbf{b} \in \mathbb{N}$ time instants is given by:

$$\mathbf{e}_{k+\mathbf{b}-1} = \mathcal{A}_{k,k+\mathbf{b}-1} \mathbf{e}_{k-1} + \mathbf{\Omega}_{k,k+\mathbf{b}-1}, \quad (32)$$

where $\mathcal{A}_{k,k+\mathbf{b}-1}$ is defined in (10) and for $\mathbf{b} \geq 2$:

$$\mathbf{\Omega}_{k,k+\mathbf{b}-1} = \sum_{\tau=k}^{k+\mathbf{b}-2} (\mathcal{A}_{\tau+1,k+\mathbf{b}-1} \mathbf{\Omega}_{\tau,\tau}) + \mathbf{\Omega}_{k+\mathbf{b}-1,k+\mathbf{b}-1}.$$

Note that $\mathbf{1}^T \mathbf{e}_k = 0$ because of (30) and $\mathbf{1}^T \mathbf{\Omega}_{k,k+\mathbf{b}-1} = 0$ because $\mathbf{1}^T \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) = 0$. Therefore, we investigate the convergence of \mathbf{e}_k along all directions that are orthogonal to $\mathbf{1}^T$. It follows from Assumption 1 that the matrix $\mathcal{A}_{k,k+\mathbf{b}-1}$ is irreducible. Therefore, the matrix $\mathcal{A}_{k,k+\mathbf{b}-1}$ is primitive [35, Lemma 8.5.4, pp. 516] and $\lambda_{N-1}(\mathcal{A}_{k,k+\mathbf{b}-1}) < 1$, where λ_{N-1} denotes the second largest eigenvalue of the matrix. Let $V_{\text{tr}} = \left[\frac{1}{\sqrt{N}} \mathbf{1}, V_s \right]$ be the orthonormal matrix of eigenvectors of the symmetric primitive matrix $\mathcal{A}_{1,\mathbf{b}}^T \mathcal{A}_{1,\mathbf{b}}$. By spectral decomposition [36], we get:

$$V_{\text{tr}}^T \mathcal{A}_{1,\mathbf{b}}^T \mathcal{A}_{1,\mathbf{b}} V_{\text{tr}} = \begin{bmatrix} \mathbf{0}^{(N-1) \times 1} & \mathbf{0}^{1 \times (N-1)} \\ V_s^T \mathcal{A}_{1,\mathbf{b}}^T \mathcal{A}_{1,\mathbf{b}} V_s & \end{bmatrix},$$

where $\frac{1}{N} \mathbf{1}^T \mathcal{A}_{1,\mathbf{b}}^T \mathcal{A}_{1,\mathbf{b}} \mathbf{1} = 1$, $\frac{1}{\sqrt{N}} \mathbf{1}^T \mathcal{A}_{1,\mathbf{b}}^T \mathcal{A}_{1,\mathbf{b}} V_s = \mathbf{0}^{1 \times (N-1)}$, and $V_s^T \mathcal{A}_{1,\mathbf{b}}^T \mathcal{A}_{1,\mathbf{b}} \frac{1}{\sqrt{N}} \mathbf{1} = \mathbf{0}^{(N-1) \times 1}$ are used. Since the eigenvectors are orthonormal, we have $V_s V_s^T + \frac{1}{N} \mathbf{1}\mathbf{1}^T = \mathbf{I}$. Left-multiplying (32) with V_s^T gives:

$$\begin{aligned}
V_s^T \mathbf{e}_{k+b-1} &= V_s^T \boldsymbol{\Omega}_{k,k+b-1} \\
&\quad + V_s^T \mathcal{A}_{k,k+b-1} (V_s V_s^T + \frac{1}{N} \mathbf{1} \mathbf{1}^T) \mathbf{e}_{k-1}, \\
&= V_s^T \boldsymbol{\Omega}_{k,k+b-1} + V_s^T \mathcal{A}_{k,k+b-1} V_s V_s^T \mathbf{e}_{k-1}. \quad (33)
\end{aligned}$$

We first investigate the stability of this system without the disturbance term $V_s^T \boldsymbol{\Omega}_{k,k+b-1}$ in (33). Let $\|V_s^T \mathbf{e}_{k+b-1}\|_2$ be a candidate Lyapunov function for this system. Therefore, we get:

$$\begin{aligned}
\|V_s^T \mathbf{e}_{k+b-1}\|_2 &= \|V_s^T \mathcal{A}_{k,k+b-1} V_s V_s^T \mathbf{e}_{k-1}\|_2 \\
&\leq \|V_s^T \mathcal{A}_{k,k+b-1} V_s\|_2 \|V_s^T \mathbf{e}_{k-1}\|_2 \\
&\leq \sigma_{\max}(\mathcal{A}_{k,k+b-1} V_s) \|V_s^T \mathbf{e}_{k-1}\|_2, \\
&= \sigma_{N-1}(\mathcal{A}_{k,k+b-1}) \|V_s^T \mathbf{e}_{k-1}\|_2,
\end{aligned}$$

where σ_{\max} denotes the largest singular value of the matrix. Since V_s^T is orthonormal (i.e., $V_s^T V_s = \mathbf{I}$) and also orthogonal to $\mathbf{1}^T$ (i.e., $V_s^T \mathbf{1} = \mathbf{0}$) and the matrix $\mathcal{A}_{k,k+b-1}^T \mathcal{A}_{k,k+b-1}$ is primitive, we have $\sigma_{\max}(\mathcal{A}_{k,k+b-1} V_s) = \sigma_{N-1}(\mathcal{A}_{k,k+b-1}) < 1$, where σ_{N-1} denotes the second largest singular value of the matrix. Therefore, the error vector $V_s^T \mathbf{e}_k$ is globally exponentially stable in absence of the disturbance term.

Since the matrix $\mathcal{A}_{k,k+b-1}$ is irreducible, the matrix $\mathcal{A}_{k,k+b(N-1)-1}$ is a positive matrix because the maximum path length between any two agents is less than or equal to $b(N-1)$ [37]. Hence the measure of irreducibility of the matrix $\mathcal{A}_{k,k+b(N-1)-1}^T \mathcal{A}_{k,k+b(N-1)-1}$ is lower bounded by $\frac{\gamma - \gamma^N}{1 - \gamma}$, and we have $\sigma_{N-1}(\mathcal{A}_{k,k+b(N-1)-1}) \leq \left(1 - \frac{4(\gamma - \gamma^N)}{(1 - \gamma)} \sin^2 \frac{\pi}{2N}\right)^{\frac{1}{2}} < 1$ [38]. Therefore, σ_m is given by (26). Moreover, it follows from Assumption 3 that $\left\| [\mathcal{L}_k^1 - \mathcal{L}_{k-1}^1, \dots, \mathcal{L}_k^N - \mathcal{L}_{k-1}^N]^T \right\|_2 \leq 2\sqrt{N}\Delta\theta_L$ because $|\mathcal{L}_k^i - \mathcal{L}_{k-1}^i| \leq 2\Delta\theta_L$. Therefore, we have:

$$\|V_s^T \boldsymbol{\Omega}_{k,k+b(N-1)-1}\|_2 \leq 2b(N-1)\sqrt{N}\Delta\theta_L.$$

Hence, in the presence of the disturbance term, we get:

$$\begin{aligned}
\|V_s^T \mathbf{e}_{k+b(N-1)-1}\|_2 &\leq \|V_s^T \boldsymbol{\Omega}_{k,k+b(N-1)-1}\|_2 \\
&\quad + \sigma_{N-1}(\mathcal{A}_{k,k+b(N-1)-1}) \|V_s^T \mathbf{e}_{k-1}\|_2, \\
&\leq \sigma_m \|V_s^T \mathbf{e}_{k-1}\|_2 + 2b(N-1)\sqrt{N}\Delta\theta_L. \quad (34)
\end{aligned}$$

Using the discrete Gronwall lemma [39, pp. 9] we obtain:

$$\begin{aligned}
\|V_s^T \mathbf{e}_k\|_2 &\leq \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor} \|V_s^T \mathbf{e}_1\|_2 \\
&\quad + \frac{1 - \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor}}{1 - \sigma_m} 2b(N-1)\sqrt{N}\Delta\theta_L. \quad (35)
\end{aligned}$$

Moreover, $\|V_s^T \mathbf{e}_1(\mathbf{x})\|_2 \leq \sqrt{N}\mathfrak{D}_1$, where \mathfrak{D}_1 is defined in (25). Therefore, it follows that for all $\mathbf{x} \in \mathcal{X}$:

$$\max_{i \in \mathcal{V}} |\mathcal{U}_k^i(\mathbf{x}) - \mathcal{L}_k^{\text{KL}}(\mathbf{x})| \leq \Xi_k, \quad \forall k \in \mathbb{N}, \quad (36)$$

$$\begin{aligned}
\text{where } \Xi_k &= \left(\sqrt{N}\mathfrak{D}_1 - \frac{2b(N-1)\sqrt{N}\Delta\theta_L}{1 - \sigma_m} \right) \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor} \\
&\quad + \frac{2b(N-1)\sqrt{N}\Delta\theta_L}{1 - \sigma_m}. \quad (37)
\end{aligned}$$

Thus, the error between the function \mathcal{U}_k^i and the function $\mathcal{L}_k^{\text{KL}}$ is bounded by Ξ_k , which depends on time instant k .

Step 2. We now show that the pdf \mathcal{T}_k^i (19) converges to the pdf \mathcal{L}_k^C (15). For all $\mathbf{x} \in \mathcal{X}$, equations (15) and (19) can be re-written as:

$$\mathcal{L}_k^C(\mathbf{x}) = N \mathcal{L}_k^{\text{KL}}(\mathbf{x}), \quad \mathcal{T}_k^i(\mathbf{x}) = N \mathcal{U}_k^i(\mathbf{x}), \quad \forall i \in \mathcal{V}.$$

Therefore, it follows from (36) that:

$$\max_{i \in \mathcal{V}} |\mathcal{T}_k^i(\mathbf{x}) - \mathcal{L}_k^C(\mathbf{x})| \leq N\Xi_k, \quad \forall k \in \mathbb{N}. \quad (38)$$

Thus, the error between the function \mathcal{T}_k^i and the function \mathcal{L}_k^C is bounded by $N\Xi_k$. We have:

$$\max_{i \in \mathcal{V}} \left| \ln \left[\frac{\mathcal{T}_k^i(\mathbf{x})}{\mathcal{T}_k^i(\boldsymbol{\psi})} \right] - \ln \left[\frac{\mathcal{L}_k^C(\mathbf{x})}{\mathcal{L}_k^C(\boldsymbol{\psi})} \right] \right| \leq N\Xi_k, \quad \forall k \in \mathbb{N}.$$

Using Lemma 1, we select $\boldsymbol{\psi} \in \mathcal{X}$ such that $\mathcal{T}_k^i(\boldsymbol{\psi}) = \mathcal{L}_k^C(\boldsymbol{\psi})$. Therefore,

$$\begin{aligned}
\max_{i \in \mathcal{V}} \left| \ln \left[\frac{\mathcal{T}_k^i(\mathbf{x})}{\mathcal{L}_k^C(\mathbf{x})} \right] \right| &\leq N\Xi_k, \quad \forall k \in \mathbb{N}, \\
e^{-N\Xi_k} &\leq \max_{i \in \mathcal{V}} \left(\frac{\mathcal{T}_k^i(\mathbf{x})}{\mathcal{L}_k^C(\mathbf{x})} \right) \leq e^{N\Xi_k}, \quad \forall k \in \mathbb{N}.
\end{aligned}$$

$$\max_{i \in \mathcal{V}} |\mathcal{T}_k^i(\mathbf{x}) - \mathcal{L}_k^C(\mathbf{x})| \leq \mathcal{L}_k^C(\mathbf{x}) (e^{N\Xi_k} - 1), \quad \forall k \in \mathbb{N}.$$

Since $\mathbf{x} \in \mathcal{X}$ can be any point, therefore:

$$\begin{aligned}
\max_{i \in \mathcal{V}} D_{L_1}(\mathcal{T}_k^i, \mathcal{L}_k^C) &= \max_{i \in \mathcal{V}} \int_{\mathcal{X}} |\mathcal{T}_k^i - \mathcal{L}_k^C| d\mu(\mathbf{x}) \\
&\leq (e^{N\Xi_k} - 1) \int_{\mathcal{X}} \mathcal{L}_k^C d\mu(\mathbf{x}) = (e^{N\Xi_k} - 1), \quad \forall k \in \mathbb{N}.
\end{aligned}$$

Hence the convergence error between the pdfs is bounded by $(e^{N\Xi_k} - 1)$.

It follows from (23)–(22) that $(e^{N\Xi_k} - 1) \leq (1 + \eta)\delta$ for all $k \geq \kappa$ and $\lim_{k \rightarrow \infty} (e^{N\Xi_k} - 1) \leq \delta$. The time step size Δ (21) is found using the steady-state error term:

$$e^{N \left(\frac{2b(N-1)\sqrt{N}\Delta\theta_L}{1 - \sigma_m} \right)} - 1 = \delta. \quad (39)$$

If $\sqrt{N}\mathfrak{D}_1 \leq \frac{2b(N-1)\sqrt{N}\Delta\theta_L}{1 - \sigma_m}$, then $(e^{N\Xi_k} - 1) \leq (1 + \eta)\delta$ for all $k \in \mathbb{N}$. Therefore, if $\mathfrak{D}_1 \leq \frac{\log(\delta+1)}{N^{\frac{3}{2}}}$, then $\kappa = 1$.

Otherwise, for $\mathfrak{D}_1 > \frac{\log(\delta+1)}{N^{\frac{3}{2}}}$, κ (24) is computed using $(e^{N\Xi_k} - 1) \leq (1 + \eta)\delta$. The constraint on TV error follows from Lemma 3. Note that our exponential-convergence proof is substantially different from the asymptotic-convergence proof in [27]. ■

Remark 1 A key advantage of the DBF algorithm is that it does not require all the sensors to observe the target. If an agent does not observe the target, then it sets its normalized likelihood function as the uniform distribution, i.e., $\mathcal{L}_k^i(\mathbf{x}) = 1$. Then this agent's likelihood function does not influence the joint likelihood function and the estimated pdfs because of the geometric nature of the fusion rule.

Moreover, the DBF algorithm avoids double counting because the summation of weights from all paths is a constant due to the weights in the adjacency matrix \mathcal{A}_k . Theorem 5 explicitly bounds the time step size Δ of the distributed estimation algorithm with the time-scale of the target dynamics. But the effectiveness of the DBF algorithm is predicated on Assumption 3. Moreover, the upper bound on the time step size Δ_{\max} (21) decreases with increasing number of agents N .

The following corollary provides sharper bounds for the special case of a static, strongly-connected communication network topology.

Corollary 6 If the communication network topology is time-invariant and strongly-connected, then the time step size Δ (21) and κ (24) in Theorem 5 are given by:

$$\Delta = \frac{(1 - \sigma_{N-1}(\mathcal{A})) \log(\delta + 1)}{2N\sqrt{N}\theta_L}, \quad (40)$$

$$\kappa = \left\lceil \frac{1}{\log \sigma_{N-1}(\mathcal{A})} \log \left(\frac{\log \left(\frac{(1+\eta)\delta+1}{\delta+1} \right)}{\log \left(\frac{e^{N^{\frac{3}{2}}\mathfrak{D}_1}}{\delta+1} \right)} \right) \right\rceil + 1, \quad (41)$$

where \mathcal{A} is the time-invariant adjacency matrix.

Proof: In this case, (34) is written as:

$$\|V_s^T \mathbf{e}_k\|_2 \leq \sigma_{N-1}(\mathcal{A}) \|V_s^T \mathbf{e}_{k-1}\|_2 + 2\sqrt{N}\Delta\theta_L.$$

Using the discrete Gronwall lemma [39, pp. 9] we obtain:

$$\begin{aligned} \|V_s^T \mathbf{e}_k\|_2 &\leq (\sigma_{N-1}(\mathcal{A}))^{k-1} \|V_s^T \mathbf{e}_1\|_2 \\ &\quad + \frac{1 - (\sigma_{N-1}(\mathcal{A}))^{k-1}}{1 - \sigma_{N-1}(\mathcal{A})} 2\sqrt{N}\Delta\theta_L. \end{aligned} \quad (42)$$

Hence, we get $\max_{i \in \mathcal{V}} |\mathcal{U}_k^i(\mathbf{x}) - \mathcal{L}_k^{\text{KL}}(\mathbf{x})| \leq \Xi_k$ for all $k \in \mathbb{N}$, where

$$\Xi_k = (\sigma_{N-1}(\mathcal{A}))^{k-1} \sqrt{N}\mathfrak{D}_1 + \frac{1 - (\sigma_{N-1}(\mathcal{A}))^{k-1}}{1 - \sigma_{N-1}(\mathcal{A})} 2\sqrt{N}\Delta\theta_L.$$

We get Δ (40) and κ (41) from $\lim_{k \rightarrow \infty} (e^{N\Xi_k} - 1) \leq \delta$ and $(e^{N\Xi_k} - 1) \leq (1 + \eta)\delta$ for all $k \geq \kappa$ respectively. ■

4.1 Robustness Analysis

In this section, we study the robustness of the DBF algorithm in the presence of communication and modeling errors. In order to implement the DBF algorithm, the agents need to communicate their estimated pdfs \mathcal{U}_{k-1}^i with their neighboring agents (see line 5 in Algorithm 1).

Remark 2 (Communication of pdfs) The information theoretic approach for communicating pdfs is studied in [40]. If particle filters are used to implement the Bayesian filter and combine the pdfs [32,41], then the resampled particles represent the agent's estimated pdf. Hence communicating pdfs is equivalent to transmitting these resampled particles. Another approach involves approximating the pdf by a weighted sum of Gaussian pdfs [42, pp. 213] and then transmitting this approximate distribution. Several techniques for estimating the Gaussian parameters are discussed in the Gaussian mixture model literature [41,43,44].

Let the pdf $\hat{\mathcal{U}}_k^i \in \Phi(\mathcal{X})$ denote the pdf \mathcal{U}_k^i that is corrupted with communication errors. Similarly, let the pdf $\hat{\mathcal{L}}_k^i \in \Phi(\mathcal{X})$ represent the normalized likelihood function \mathcal{L}_k^i that is corrupted with modeling errors. We first state the assumptions on these errors and then state the main result of this section.

Assumption 4 There exists time-invariant constants $\varepsilon_U \geq 0$ and $\varepsilon_L \geq 0$ such that for all agents $i \in \mathcal{V}$:

$$e^{-\varepsilon_U} \leq \frac{\hat{\mathcal{U}}_k^i(\mathbf{x})}{\mathcal{U}_k^i(\mathbf{x})} \leq e^{\varepsilon_U}, \quad \forall \mathbf{x} \in \mathcal{X}, \forall k \in \mathbb{N}, \quad (43)$$

$$e^{-\varepsilon_L} \leq \frac{\hat{\mathcal{L}}_k^i(\mathbf{x})}{\mathcal{L}_k^i(\mathbf{x})} \leq e^{\varepsilon_L}, \quad \forall \mathbf{x} \in \mathcal{X}, \forall k \in \mathbb{N}. \quad (44)$$

Therefore, $|\mathcal{U}_k^i - \hat{\mathcal{U}}_k^i| \leq 2\varepsilon_U$ and $|\mathcal{L}_k^i - \hat{\mathcal{L}}_k^i| \leq 2\varepsilon_L$, where $\mathcal{U}_k^i(\mathbf{x}) = \ln \left[\frac{\mathcal{U}_k^i(\mathbf{x})}{\mathcal{U}_k^i(\psi)} \right]$ and $\mathcal{L}_k^i(\mathbf{x}) = \ln \left[\frac{\mathcal{L}_k^i(\mathbf{x})}{\mathcal{L}_k^i(\psi)} \right]$.

Corollary 7 Under Assumptions 1–4, the time step size Δ (21) in Theorem 5 is given by:

$$\Delta = \frac{(1 - \sigma_m) \log(\delta + 1)}{2bN(N-1)\sqrt{N}\theta_L} - \frac{2\varepsilon_L + \varepsilon_U}{\theta_L}, \quad (45)$$

where ε_U and ε_L are defined in Assumption 4.

Proof: Equation (18) can be written as:

$$\mathcal{U}_k^i = \begin{cases} \hat{\mathcal{L}}_1^i & \text{if } k = 1 \\ \sum_{j=1, j \neq i}^N \mathcal{A}_k[i, j] \mathcal{U}_{k-1}^j + \mathcal{A}_k[i, i] \mathcal{U}_{k-1}^i + \hat{\mathcal{L}}_k^i - \hat{\mathcal{L}}_{k-1}^i & \text{if } k \geq 2 \end{cases}, \quad (46)$$

Substituting the bounds from Assumption 4 gives:

$$|\mathcal{W}_1^i - \mathcal{L}_1^i| \leq 2\varepsilon_L,$$

$$|\mathcal{W}_k^i - \sum_{j=1}^N \mathcal{A}_k[i, j] \mathcal{W}_{k-1}^j - \mathcal{L}_k^i + \mathcal{L}_{k-1}^i| \leq 2\varepsilon_U + 4\varepsilon_L.$$

The evolution of the error vector \mathbf{e}_k is now given by:

$$\mathbf{e}_k = \mathcal{A}_k \mathbf{e}_{k-1} + \hat{\mathbf{\Omega}}_{k,k}, \quad \forall k \geq 2, \quad (47)$$

where $\|\hat{\mathbf{\Omega}}_{k,k}\|_2 \leq \|\mathbf{\Omega}_{k,k}\|_2 + 2\sqrt{N}(\varepsilon_U + 2\varepsilon_L)$.

Similar to the proof of Theorem 5, we get:

$$\|V_s^T \mathbf{e}_k\|_2 \leq \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor} \|V_s^T \mathbf{e}_1\|_2$$

$$+ \frac{1 - \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor}}{1 - \sigma_m} 2b(N-1)\sqrt{N}(\Delta\theta_L + 2\varepsilon_L + \varepsilon_U).$$

Hence, we get $\max_{i \in \mathcal{V}} |\mathcal{W}_k^i(\mathbf{x}) - \mathcal{L}_k^{\text{KL}}(\mathbf{x})| \leq \Xi_k$ for all $k \in \mathbb{N}$, where

$$\Xi_k = \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor} \sqrt{N} \mathfrak{D}_1$$

$$+ \frac{1 - \sigma_m^{\lfloor \frac{k-1}{b(N-1)} \rfloor}}{1 - \sigma_m} 2b(N-1)\sqrt{N}(\Delta\theta_L + 2\varepsilon_L + \varepsilon_U).$$

We get Δ (45) and κ (24) from $\lim_{k \rightarrow \infty} (e^{N\Xi_k} - 1) \leq \delta$ and $(e^{N\Xi_k} - 1) \leq (1 + \eta)\delta$ for all $k \geq \kappa$ respectively. ■

It follows from Corollary 7 that in order to generate satisfactory estimates using the DBF algorithm, the bounds $\varepsilon_U, \varepsilon_L$ should be substantially smaller than δ .

4.2 Special Case: DBF-Kalman Information Filter

In this section, we apply the DBF algorithm to the special case where the target dynamics and measurement models are given by linear systems with additive Gaussian noise:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k, \quad \forall k \in \mathbb{N}, \quad (51)$$

$$\mathbf{y}_k^i = \mathbf{H}_k^i \mathbf{x}_k + \mathbf{v}_k^i, \quad \forall k \in \mathbb{N}, \forall i \in \mathcal{V}, \quad (52)$$

where the process noise $\mathbf{w}_k = \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ and the measurement noise $\mathbf{v}_k^i = \mathcal{N}(\mathbf{0}, \mathbf{R}_k^i)$ are zero mean multivariate normal distributions.

Therefore, we adopt the information filter-based representation [45,46]. The pseudo-code of the DBF-Kalman information filtering algorithm for linear-Gaussian models is given in Algorithm 2. The prior pdf $\mathcal{S}_k^i = \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)$, the posterior pdf $\mathcal{W}_k^i = \mathcal{N}(\hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i)$, and the estimated pdfs $\mathcal{U}_k^i = \mathcal{N}((\mathbf{U}_k^i)^{-1} \mathbf{u}_k^i, (\mathbf{U}_k^i)^{-1})$, $\mathcal{T}_k^i = \mathcal{N}((\mathbf{T}_k^i)^{-1} \mathbf{t}_k^i, (\mathbf{T}_k^i)^{-1})$ are also multivariate normal distributions.

Algorithm 2. DBF-Kalman Information Filtering Algorithm

1. (i^{th} agent's steps at k^{th} time instant)
2. Compute the prior pdf $\mathcal{S}_k^i = \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)$

$$\hat{\mathbf{z}}_{k-1|k-1}^i = (\mathbf{P}_{k-1|k-1}^i)^{-1} \hat{\mathbf{x}}_{k-1|k-1}^i,$$

$$\mathbf{Z}_{k-1|k-1}^i = (\mathbf{P}_{k-1|k-1}^i)^{-1},$$

$$\mathbf{M}_{k-1}^i = (\mathbf{F}_{k-1}^{-1})^T \mathbf{Z}_{k-1|k-1}^i \mathbf{F}_{k-1}^{-1},$$

$$\mathbf{Z}_{k|k-1}^i = (\mathbf{I} - \mathbf{M}_{k-1}^i (\mathbf{M}_{k-1}^i + \mathbf{Q}_{k-1}^{-1})^{-1}) \mathbf{M}_{k-1}^i,$$

$$\hat{\mathbf{z}}_{k|k-1}^i = (\mathbf{I} - \mathbf{M}_{k-1}^i (\mathbf{M}_{k-1}^i + \mathbf{Q}_{k-1}^{-1})^{-1}) (\mathbf{F}_{k-1}^{-1})^T \hat{\mathbf{z}}_{k-1|k-1}^i,$$

$$\mathbf{P}_{k|k-1}^i = (\mathbf{Z}_{k|k-1}^i)^{-1}, \quad (48)$$

$$\hat{\mathbf{x}}_{k|k-1}^i = \mathbf{P}_{k|k-1}^i \hat{\mathbf{z}}_{k|k-1}^i. \quad (49)$$
3. Obtain local measurement \mathbf{y}_k^i
4. Receive pdfs \mathcal{U}_{k-1}^j from agents $j \in \mathcal{J}_k^i$
5. Compute the pdfs \mathcal{U}_k^i and \mathcal{T}_k^i as follows:
$$\mathbf{i}_k^i = (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{y}_k^i,$$

$$\mathbf{I}_k^i = (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i,$$

$$\mathbf{u}_k^i = \begin{cases} \mathbf{i}_1^i & \text{if } k = 1 \\ \mathbf{i}_k^i - \mathbf{i}_{k-1}^i + \sum_{j \in \mathcal{J}_k^i} \mathcal{A}_k[i, j] \mathbf{u}_{k-1}^j, & \text{if } k \geq 2 \end{cases},$$

$$\mathbf{U}_k^i = \begin{cases} \mathbf{I}_1^i & \text{if } k = 1 \\ \mathbf{I}_k^i - \mathbf{I}_{k-1}^i + \sum_{j \in \mathcal{J}_k^i} \mathcal{A}_k[i, j] \mathbf{U}_{k-1}^j, & \text{if } k \geq 2 \end{cases},$$

$$\mathbf{t}_k^i = N \mathbf{u}_k^i, \quad \mathbf{T}_k^i = N \mathbf{U}_k^i,$$
6. Compute the posterior pdf $\mathcal{W}_k^i = \mathcal{N}(\hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i)$

$$\hat{\mathbf{z}}_{k|k}^i = \hat{\mathbf{z}}_{k|k-1}^i + \mathbf{t}_k^i, \quad \mathbf{Z}_{k|k}^i = \mathbf{Z}_{k|k-1}^i + \mathbf{T}_k^i,$$

$$\mathbf{P}_{k|k}^i = (\mathbf{Z}_{k|k}^i)^{-1}, \quad \hat{\mathbf{x}}_{k|k}^i = \mathbf{P}_{k|k}^i \hat{\mathbf{z}}_{k|k}^i. \quad (50)$$

4.3 Special Case: Multiple Consensus Loops within Each Time Instant

In this section, we show that the proposed DBF algorithm can be easily extended to recursively combine local likelihood functions using multiple consensus loops within each time instant so that each agent's estimated likelihood function converges to the joint likelihood function \mathcal{L}_k^C (15). Then, the resultant DBF algorithm is equivalent to the Bayesian consensus algorithms in [14,15]. Note that multiple consensus loops within each time step significantly reduces the practicality of such algorithms. Let the pdfs $\mathcal{U}_{k,\nu}^i \in \Phi(\mathcal{X})$ and $\mathcal{T}_{k,\nu}^i \in \Phi(\mathcal{X})$ denote to the local pdfs of the i^{th} agent during the ν^{th} consensus loop at the k^{th} time instant. Since the pdf \mathcal{L}_k^i is not updated during the k^{th} time instant, we define the pdfs $\mathcal{L}_{k,\nu}^i = \mathcal{L}_k^i$ for all $\nu \in \mathbb{N}$. During the ν^{th} consensus loop, each agent updates its local pdfs $\mathcal{U}_{k,\nu}^i$ and $\mathcal{T}_{k,\nu}^i$ using the following fusion rule:

$$\mathcal{U}_{k,\nu}^i = \begin{cases} \mathcal{L}_{k,1}^i & \text{if } \nu = 1 \\ \frac{\prod_{j \in \mathcal{J}_k^i} (\mathcal{U}_{k,\nu-1}^j)^{\mathcal{A}_k[i,j]}}{\int_{\mathcal{X}} \prod_{j \in \mathcal{J}_k^i} (\mathcal{U}_{k,\nu-1}^j)^{\mathcal{A}_k[i,j]} d\mu(\mathbf{x})} & \text{if } \nu \geq 2 \end{cases}, \quad (53)$$

$$\mathcal{T}_k^i = \frac{(\mathcal{U}_{k,\nu}^i)^N}{\int_{\mathcal{X}} (\mathcal{U}_{k,\nu}^i)^N d\mu(\mathbf{x})}. \quad (54)$$

Theorem 8 [17,18] Assuming \mathcal{G}_k is strongly connected, each agent's pdf $\mathcal{T}_{k,\nu}^i$ globally exponentially converges pointwise to \mathcal{L}_k^C (15). After n_{loop} consensus loops, the ℓ_2 norm of the error vector $\mathbf{e}_{k,\nu} := [D_{L_1}(\mathcal{T}_{k,\nu}^1, \mathcal{L}_k^C), \dots, D_{L_1}(\mathcal{T}_{k,\nu}^N, \mathcal{L}_k^C)]^T$ is bounded by $\|\mathbf{e}_{k,n_{\text{loop}}}\|_2 \leq (\sigma_{N-1}(\mathcal{A}_k))^{(n_{\text{loop}}-1)} 2\sqrt{N}$.

The proof follows from Theorem 2 and 4 in [17]. Thus, the distributed estimation algorithm in [14,15] is a special case of our DBF algorithm.

5 Numerical Simulations

In this section, we demonstrate the properties of the DBF algorithm using a benchmark example in Section 5.1 and a complex multi-agent estimation and control task in Section 5.2.

5.1 Benchmark Example

In this subsection, we compare the performance of the DBF algorithms with the centralized multi-sensor Bayesian filtering algorithms using the benchmark example studied in [6,16,47]. The target dynamics is modeled by a linear (nearly constant velocity) model:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k, \text{ where } \mathbf{Q} = \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 \\ \frac{\Delta^2}{2} & \Delta & 0 & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ 0 & 0 & \frac{\Delta^2}{2} & \Delta \end{bmatrix}$$

is the covariance matrix of the process noise \mathbf{w}_k , Δ is the time step size, and the state vector \mathbf{x}_k denotes the position and velocity components along the coordinate axes, i.e., $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$. As shown in Fig. 3, 50 sensing agents are distributed over the given region and are able to communicate with their neighboring agents. The undirected communication network topology is assumed to be time-invariant. Local-degree weights are used to compute the doubly stochastic adjacency matrix \mathcal{A}_k as:

$$\mathcal{A}_k[i, j] = \frac{1}{\max(d_i, d_j)}, \quad \forall j \in \mathcal{J}_k^i \text{ and } i \neq j, \\ \mathcal{A}_k[i, i] = 1 - \sum_{j \in \mathcal{V} \setminus \{i\}} \mathcal{A}_k[i, j],$$

where d_i denotes the degree of the i^{th} agent.

In Scenario 1, five of these agents are equipped with non-linear position sensors that can measure their distance to the target using Time of Arrival (TOA) sensors. Another five agents are equipped with Direction of Arrival (DOA) sensors that can measure the bearing angle between the target and themselves. The remaining agents

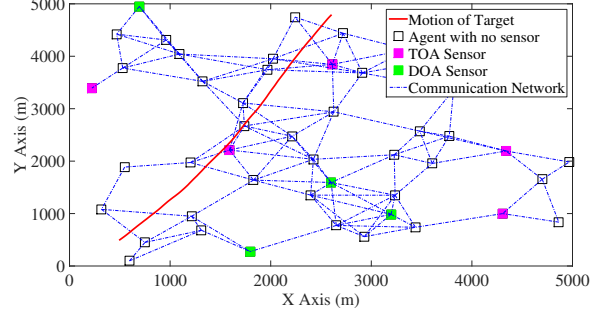


Fig. 3. The motion of the target, the position of sensing agents (5 TOA sensors, 5 DOA sensors, and 40 agents with no sensors), and their communication network topology are shown.

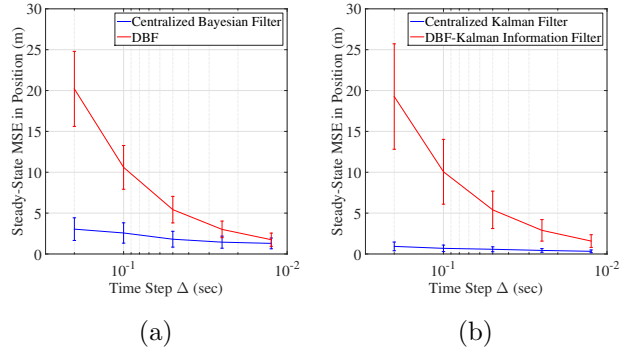


Fig. 4. Variation of steady-state MSE in position with respect to time step size Δ is shown for (a) the centralized Bayesian filtering algorithm and the DBF algorithm in Scenario 1 and (b) the centralized Kalman filtering algorithm and the DBF algorithm for linear-Gaussian models in Scenario 2.

do not have any sensors. The measurement models for these sensors are given by:

$$\mathbf{h}_k^i(\mathbf{x}_k, \mathbf{v}_k^i) = \begin{cases} \text{atan2}(x_k - x^i, y_k - y^i) + \mathbf{v}_{k,DOA}^i & \text{for DOA sensor} \\ \sqrt{(x_k - x^i)^2 + (y_k - y^i)^2} + \mathbf{v}_{k,TOA}^i & \text{for TOA sensor} \end{cases}, \quad (55)$$

where (x^i, y^i) denotes the position of the i^{th} agent and atan2 is the 4-quadrant inverse tangent function. The DOA sensor's measurement noise $\mathbf{v}_{k,DOA}^i = \mathcal{N}(0, \sigma_\theta)$ has variance $\sigma_\theta = 2^\circ$ and the TOA sensor's measurement noise $\mathbf{v}_{k,TOA}^i = \mathcal{N}(0, \sigma_r)$ has variance $\sigma_r = 10$ m.

In Scenario 1, each agent executes the DBF algorithm in Algorithm 1 using particle filters [48] with 10^4 particles. The comparison between the DBF algorithm and the centralized Bayesian filtering algorithm for varying time step sizes (Δ) is shown in Fig. 4(a). The same target motion, shown in Fig. 3, is used for all simulations. We see that the DBF algorithm's steady-state mean-square-error (MSE) in position converges to that of the centralized algorithm as the time step size Δ decreases (i.e., the steady-state MSE is smaller than 5 m if the time step size $\Delta \leq 0.05$ sec). Note that the MSE of the

centralized algorithm does not change much with time step size because it is constrained by the measurement noise intensities. This shows that the performance of the DBF algorithm approaches the performance of the centralized Bayesian filter as the time step size is reduced. Moreover, Fig. 5 shows that the L_1 distances between the estimated likelihood functions and the joint likelihood function are bounded by δ .

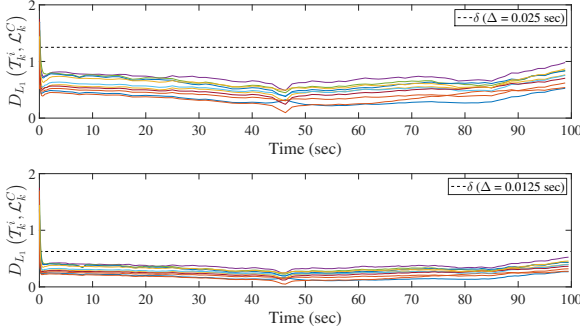


Fig. 5. The trajectories of the L_1 distances between the estimated likelihood functions and the joint likelihood function for the ten sensing agents are shown.

In Scenario 2, the same ten agents (having DOA or TOA sensors) have linear position sensors $\mathbf{h}_k^i(\mathbf{x}_k, \mathbf{v}_k^i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_{k,lin}^i$, with measurement noise $\mathbf{v}_{k,lin}^i = \mathcal{N}(\mathbf{0}, R_k^i)$ and covariance matrix $R_k^i = 15\mathbf{I}$. Here, each agent executes the DBF-Kalman information filtering algorithm from Algorithm 2. Fig. 4(b) shows that the performance of the DBF-Kalman information filtering algorithm approaches the performance of the centralized Kalman filtering algorithm as the time step size is reduced.

5.2 Multi-agent Relative Position Estimation for Formation Control

In this subsection, N agents estimate their relative positions using only range measurements, and then reconfigure to a N -sided regular polygon. Specifically, each agent can only measure the distance to its nearest two neighbors using a TOA sensor, whose measurement model is described in (55). Each agent simultaneously executes N DBF algorithms to estimate the relative positions of all the agents. The i^{th} agent's dynamics and control inputs are given by:

$$\begin{aligned} \mathbf{x}_{k+1}^i &= \mathbf{x}_k^i + \Delta \mathbf{u}_k^i, \\ \mathbf{u}_k^i &= \sum_{j \in \mathcal{N}_k^i} APF(\hat{\mathbf{x}}_k^{i,j}, \hat{\mathbf{x}}_k^{i,i}, d) + APF(\hat{\mathbf{x}}_k^{i,CM}, \hat{\mathbf{x}}_k^{i,i}, d_{CM}) \end{aligned}$$

where \mathcal{N}_k^i denotes the two nearest neighbors of the i^{th} agent and $\hat{\mathbf{x}}_k^{i,j}$ is the i^{th} agent's estimate of the j^{th} agent's position, which is obtained using the DBF algorithms. The agents use the artificial potential field (APF) based

approach to maintain a distance d from their nearest neighbors, i.e.:

$$APF(\hat{\mathbf{x}}_k^{i,j}, \hat{\mathbf{x}}_k^{i,i}, d) = \frac{(\hat{\mathbf{x}}_k^{i,j} - \hat{\mathbf{x}}_k^{i,i})}{r_k^{i,j}} \left(a r_k^{i,j} - \frac{a d^2}{r_k^{i,j}} \right),$$

where $r_k^{i,j} = \|\hat{\mathbf{x}}_k^{i,j} - \hat{\mathbf{x}}_k^{i,i}\|_2$, and maintain a distance $d_{CM} = \frac{d}{2 \cos(\frac{\pi}{2} - \frac{\pi}{N})}$ from the estimated center of mass $\hat{\mathbf{x}}_k^{i,CM} = \frac{1}{N} \sum_{j=1}^N \hat{\mathbf{x}}_k^{i,j}$. In the propagation step of the DBF algorithm, the agents use their estimated positions to estimate the control input applied by other agents. Therefore, the estimation errors contribute to the process noise in the propagation step. During the fusion step at k^{th} time instant, the i^{th} agent communicates with the j^{th} agent if either $j \in \mathcal{N}_k^i$ or $i \in \mathcal{N}_k^j$.

In these simulations, we use $a = 0.1$, $d = 1$ m, $\Delta = 0.1$ sec, and 10^3 particles to execute each DBF algorithm. At the start of the estimation process, the particles are selected from a uniform distribution over the state space $\mathcal{X} = [-N, N] \times [-N, N]$. The simulation results for multiple values of N are shown in Fig. 6. Since the agents only use relative measurements, the orientation of the final N -sided regular polygon in the global frame is not fixed. Therefore, we conclude that the N agents successfully estimate their relative positions using the DBF algorithms and achieve the complex desired formations.

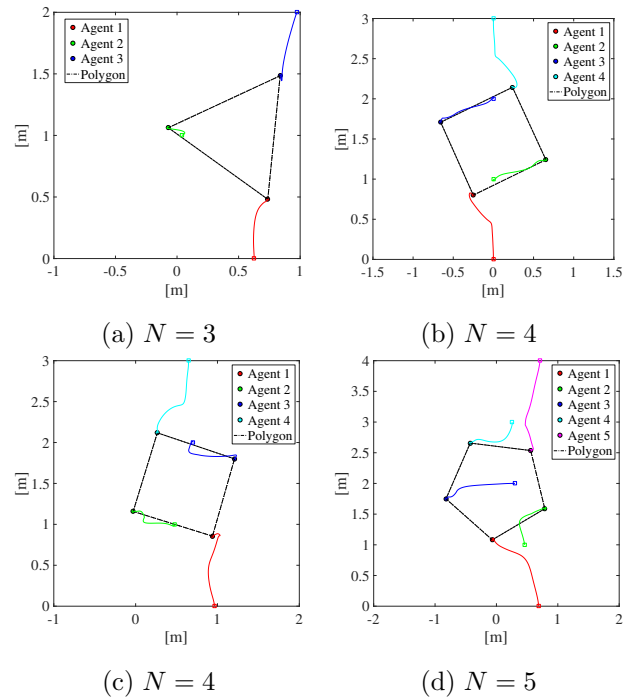


Fig. 6. The initial position (\square), the final position (\circ), the trajectories of all the agents, and the final regular polygon are shown for $N = 3, 4, 5$ agents.

6 Conclusions

In this paper, we presented a novel, discrete-time distributed estimation algorithm, namely the DBF algorithm, that ensures that each agent's estimated likelihood function converges to an error ball around the joint likelihood function of the centralized multi-sensor Bayesian filtering algorithm. We have rigorously proven the convergence properties of this algorithm. We have shown an explicit connection between the time step size of the distributed estimation algorithm and the time-scale of the target dynamics. We also presented the DBF-Kalman information filtering algorithm for the special case of linear-Gaussian models. The properties of these algorithms are illustrated using complex numerical examples. We envisage that the novel proof techniques presented in this paper can also be used in other distributed estimation algorithms which rely on the LogOP scheme.

References

- [1] J. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Autom. Control*, vol. 24, no. 2, pp. 266–269, 1979.
- [2] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," *IEEE Trans. Autom. Control*, vol. 27, no. 3, pp. 650 – 655, 1982.
- [3] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: covariance intersection revisited," *IEEE Trans. Autom. Control*, vol. 47, no. 11, pp. 1879–1882, 2002.
- [4] M. Kamgarpour and C. Tomlin, "Convergence properties of a decentralized Kalman filter," in *IEEE Conf. Decision Control*, pp. 3205–3210, IEEE, 2008.
- [5] R. Olfati-Saber, "Kalman-consensus filter : Optimality, stability, and performance," in *IEEE Conf. Decision Control*, (Shanghai, China), pp. 7036–7042, December 2009.
- [6] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1410–1415, 2015.
- [7] M. Rashedi, J. Liu, and B. Huang, "Communication delays and data losses in distributed adaptive high-gain ekf," *AICHE Journal*, vol. 62, no. 12, pp. 4321–4333, 2016.
- [8] G. Pavlin, P. Oude, M. Maris, J. Nunnink, and T. Hood, "A multi-agent systems approach to distributed Bayesian information fusion," *Inform. Fusion*, vol. 11, pp. 267–282, 2010.
- [9] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Games and Economic Behavior*, vol. 76, pp. 210–225, 2012.
- [10] A. Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *arXiv preprint arXiv:1508.05161*, 2015.
- [11] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-Gaussian dependence," in *Int. Conf. Information Fusion*, (Singapore), pp. 1876–1883, July 2012.
- [12] N. Ahmed, J. Schoenberg, and M. Campbell, "Fast weighted exponential product rules for robust general multi-robot data fusion," in *Robotics: Science and Systems VIII* (N. Roy, P. Newman, and S. Srinivasa, eds.), pp. 9–16, MIT Press, 2013.
- [13] C. S. R. Fraser, L. F. Bertuccelli, H.-L. Choi, and J. P. How, "A hyperparameter consensus method for agreement under uncertainty," *Automatica*, vol. 48, no. 2, pp. 374 – 380, 2012.
- [14] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, 2012.
- [15] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Consensus-based distributed particle filtering with distributed proposal adaptation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3029–3041, 2014.
- [16] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.
- [17] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using Bayesian consensus filtering," in *Proc. Amer. Control Conf.*, (Portland, OR), pp. 634–641, June 2014.
- [18] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using Bayesian consensus filtering." <https://arxiv.org/abs/1403.3117>.
- [19] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statistical Assoc.*, vol. 69, no. 345, pp. 688 – 704, 1960.
- [20] M. Bacharach, "Normal Bayesian dialogues," *J. Amer. Statistical Assoc.*, vol. 74, no. 368, pp. 837 – 846, 1979.
- [21] S. French, "Consensus of opinion," *European J. Operational Research*, vol. 7, pp. 332 – 340, 1981.
- [22] C. Genest and J. V. Zidek, "Combining probability distributions: A critique and an annotated bibliography," *Statistical Sci.*, vol. 1, no. 1, pp. 114 – 135, 1986.
- [23] G. L. Gilardoni and M. K. Clayton, "On reaching a consensus using DeGroot's iterative pooling," *Ann. Stat.*, vol. 21, no. 1, pp. 391 – 401, 1993.
- [24] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520 – 1533, 2004.
- [25] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988 – 1001, 2003.
- [26] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.
- [27] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.
- [28] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.
- [29] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
- [30] H. Durrant-Whyte and T. C. Henderson, *Springer Handbook of Robotics*, ch. Multisensor Data Fusion, pp. 585–610. Springer, 2008.
- [31] N. Gordon, B. Ristic, and S. Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, London, 2004.

- [32] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, February 2002.
- [33] O. Lebeltel, P. Bessiere, J. Diard, and E. Mazer, "Bayesian robot programming," *Autonomous Robots*, vol. 16, pp. 49–79, January 2004.
- [34] M.-H. Chen, "Bayesian computation: From posterior densities to Bayes factors, marginal likelihoods, and posterior model probabilities," in *Bayesian Thinking, Modeling and Computation* (D. K. Dey and C. R. Rao, eds.), Handbook of Statistics, ch. 15, pp. 437 – 457, Amsterdam: Elsevier, 2005.
- [35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, England: Cambridge University Press, 1985.
- [36] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, "Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs," *Automatica*, vol. 49, pp. 1148–1161, May 2013.
- [37] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic and distributed control of a large-scale swarm of autonomous agents," *IEEE Trans. Robotics*, vol. 33, pp. 1103–1123, Oct. 2017.
- [38] M. Fiedler, "Bounds for eigenvalues of doubly stochastic matrices," *Linear Algebra and Its Applications*, vol. 5, no. 3, pp. 299–310, 1972.
- [39] A. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis*. Cambridge, England: Cambridge University Press, 1998.
- [40] G. Kramer and S. A. Savari, "Communicating probability distributions," *IEEE Trans. Inf. Theory*, vol. 53, pp. 518–525, February 2007.
- [41] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, pp. 2602–2612, Oct. 2003.
- [42] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Mineola, New York: Dover Publications, 2005.
- [43] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. New York, N.Y.: M. Dekker, 1988.
- [44] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometric Recognition*, February 2008.
- [45] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC press, 1998.
- [46] H. Fourati, *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*. Series: Devices, Circuits, and Systems, CRC Press, Taylor & Francis Group LLC, 2015.
- [47] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2004.
- [48] D. A. A. Marín, "Particle filter tutorial." (<http://www.mathworks.com/matlabcentral/fileexchange/35468-particle-filter-tutorial>) [retrieved on 13 December 2017].
- [49] R. Durrett, *Probability: Theory and Examples*. Thomson Brooks, 2005.
- [50] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Soc., 2009.

A Proof of Lemma 1

If this claim is untrue, then either $0 < \mathcal{P}(\mathbf{x}) < \mathcal{Q}(\mathbf{x})$ or $0 < \mathcal{Q}(\mathbf{x}) < \mathcal{P}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$. Hence either $\int_{\mathcal{X}} \mathcal{P}(\mathbf{x}) d\mu(\mathbf{x}) = 1 < \int_{\mathcal{X}} \mathcal{Q}(\mathbf{x}) d\mu(\mathbf{x})$ or $\int_{\mathcal{X}} \mathcal{Q}(\mathbf{x}) d\mu(\mathbf{x}) < \int_{\mathcal{X}} \mathcal{P}(\mathbf{x}) d\mu(\mathbf{x}) = 1$, which results in contradiction since $\int_{\mathcal{X}} \mathcal{Q}(\mathbf{x}) d\mu(\mathbf{x}) = 1$. Hence, such a $\psi \in \mathcal{X}$ must exist.

B Proof of Lemma 2

Since $\lim_{k \rightarrow \infty} \mathcal{P}_k^i(\mathbf{x}) = \mathcal{P}^*(\mathbf{x})$, we have $\lim_{k \rightarrow \infty} (\ln \mathcal{P}_k^i(\mathbf{x}) - \ln \mathcal{P}_k^i(\psi)) = \ln \mathcal{P}^*(\mathbf{x}) - \ln \mathcal{P}^*(\psi)$. From Lemma 1, substituting $\lim_{k \rightarrow \infty} \mathcal{P}_k^i(\psi) = \mathcal{P}^*(\psi)$ gives $\lim_{k \rightarrow \infty} \mathcal{P}_k^j(\mathbf{x}) = \mathcal{P}^*(\mathbf{x})$ since logarithm is a monotonic function.

C Proof of Lemma 3

It follows from Scheffé's theorem [49, pp. 84] that if the pdfs converge pointwise, then their induced measures converge in TV. The relationship between TV error and L_1 distance follows from [50, pp. 48].